# Release Notes for Katana 2.0v1b2

## Release Date

29 October 2014

## System Requirements

- Katana 2.0v1b2 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

### Supported Renderers

Katana 2.0v1b2 supports PRMan 17.0 and Arnold 4.1.3.3. The supplied renderer plug-ins are compiled and tested against these versions, using GCC 4.1.2. Minor version increments of PRMan and Arnold may work, as long as they are API compliant with the supported versions.

Support for PRMan 18.0 and above is provided directly by Pixar using their RenderMan Studio for Katana plug-in. To get access to RenderMan Studio for Katana please contact Pixar directly.

To use a version of PRMan or Arnold other than those listed above, you may need to recompile the renderer plug-in.

To expose new features and options, you may need to modify the renderer plug-in.

Using a version of PRMan or Arnold other than those listed above may produce unexpected behavior. Using a compiler other than GCC 4.1.2 may produce unexpected behavior. Please note that we can only guarantee to respond to Katana bug reports when they are reproducible with the supplied versions of the renderer plug-in, compiled with the supported version of GCC.

## Third-Party Dependencies

Katana 2.0v1b2 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0.a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

# Feature Enhancements and Important Changes for 2.0v1b2

- BUG ID 44641 - Shader args file search paths were incorrect for shaders selected by browsing to a shader file, and shaders in sub-directories of shader paths (where the shader name is reported as containing **/** characters).
- BUG ID 45631 - ShadingNode material parameters were populated in the UI prior to receiving paging hints, which could be an expensive operation.
- BUG ID 45465 - Global light linking information is now stored in Look Files using LookFileBake and automatically retrieved through LookFileLightAndConstraintActivator nodes.

# Bug Fixes

- BUG ID 42670 - If the **Scene Graph** tab had keyboard focus, certain keyboard shortcuts did not function, notably **P** for **Render** > **Render View Node** and **** for **Render** > **Repeat Previous Render**.
- BUG ID 43369 - Canceling a render orphaned any child Python interpreter processes (spawned by the renderboot process only when required by Python Asset plug-ins or AttributeScript nodes) and only terminated them when Katana was closed.

- BUG ID 43734 - The **Render** > **Live Rendering** > **Update Live Render** menu command did not perform a scene graph cook when the **3D Update Mode** was set to **Manual**. The menu command has been removed. The **Trigger 3D Update** button in the **Monitor** tab's toolbar should be used instead.
- BUG ID 44059 - When using a BoundsAdjust node to pad bounds, ancestor bounds were not adjusted if its **padAmount.when** parameter was set to **deferred**.
- BUG ID 44079 - Information about material assignment on ancestor locations was not displayed in the **Attributes** tab.
- BUG ID 44101- The PrimitiveCreate node's **brick map** type created primitive locations with type **brick map** as opposed to **brickmap**.
- BUG ID 44333 - The default value for **shutterClose** in Katana's graph state had incorrectly changed from **1.0** to **0.0** when returned by the **NodegraphAPI.GetCurrentGraphState().getShutterClose()** function, and when set in the **scenegraphGenerator.args.system.timeSlice.shutterClose** attribute of locations of type **scenegraph generator** that are created by ScenegraphGeneratorSetup nodes.
- BUG ID 44520 - Katana crashed when evaluating attributes on locations not under **/root** location.
- BUG ID 44537 -The help text of the **addToConstraintList** parameter in constraint nodes, such as AimConstraint and PointConstraint, was inaccurate.
- BUG ID 44565 - The GafferThree item delegates used for the **Name**, **Mute,** and **Solo** columns were expensive to draw, causing a delay in the display of the UI.
- BUG ID 44670 - Restoring a previously renamed root location on a GafferThree node resulted in an error being raised.
- BUG ID 44827 - Various issues related to the display of animated proxies in the **Viewer** tab have been resolved, including the Viewer sometimes combining proxy geometry for multiple locations.

  See Bug 45963 in the Known Issues section for details on a remaining bug.
- BUG ID 45315 - When adding an input port to a VariableSwitch node in the Node Graph tab, no matching pattern in its pattern parameter was created.
- BUG ID 45348 - Keys like **F2**, **Enter,** or **Return** incorrectly allowed users to temporarily change the names of locations displayed in the **Scene Graph** tab.
- BUG ID 45354 - Shader parameters whose values are arrays of variable length were not supported in Katana 2.0.

  Error handling for the processing of shader parameters from renderer info plug-ins has been improved by checking for invalid default attributes.

  When using **FnAttribute::DataBuilder**-based attribute builders, such as **IntBuilder** and **FloatBuilder**, builders to which no values have been added now create an empty array attribute of the respective type, instead of an invalid attribute.
- BUG ID 45385 - When setting the **variableName** parameter of a VariableSet node to an invalid name, for example one containing dots (**.**), and clicking the **Graph State Variables** button in the **Parameters** tab, an exception was raised.
- BUG ID 45477 - When dragging and dropping text within a text edit box parameter widget, the entire text of the parameter was replaced.
- BUG ID 45488 - When the **Dim Nodes Unconnected to Viewed Node** feature of the **Node Graph** tab was turned on, connected nodes inside a Group, LiveGroup, or SuperTool-based node were incorrectly dimmed.

- BUG ID 45630 - When registering a custom layered menu with a custom keyboard shortcut in the **Node Graph** tab, and opening the tab's **Find** pop-up, pressing the keyboard shortcut of the layered menu incorrectly triggered the layered menu instead of inserting the character that corresponded to the pressed key into the text edit widget.
- BUG ID 45681- When using RfK 19 beta 4, Katana crashed when exiting. Furthermore, all PRMan renders also terminated with a ”render unsuccessful” message in render logs. This issue was caused by improper symbol visibility in one of Katana's shared libraries.
- BUG ID 45684 - The **Catalog** tab, when containing a large number of items, was slow to refresh upon switching view or deleting items.
- BUG ID 45721- Attribute widget hints were not correctly updated in the **Attributes** tab when switching between different locations having the same attribute values.
- BUG ID 45727 - Material widget hints in Look Files baked from Katana 1.x were not supported (not being automatically upgraded from a Python dictionary to a **GroupAttribute**). See Bug 46148 in the Known Issues section for details of remaining unimplemented functionality.

  Additionally, resolving Look Files with an explicit inclusion of **/root/materials** as a selected material tree created extraneous scene graph hierarchy.
- BUG ID 45799 - The **shaders** parent of the **parameters** group parameter, which should be hidden in the Material node's UI, was displayed for network materials.
- BUG ID 45944 - When a Katana documentation PDF is viewed in Firefox, some headings displayed incorrectly due to missing fonts.
- BUG ID 45962 - The **Node Graph** tab's **Find** pop-up incorrectly listed deleted nodes.
- BUG ID 45978 - The **FnViewerModifierInput::getDisplayMatrix()** method used in Viewer Modifier Plug-ins (VMP) incorrectly returned an invalid **DoubleAttribute**.
- BUG ID 46128 - The RenderSettings node’s **cropWindow** parameter was not respected in renders.

# Known Issues

## Issues Specific to 2.0

- BUG ID 46175 - ShadingNode: Hints for parameters within a page may be ignored until the page group is closed and reopened.
- BUG ID 46148 - Material widget hints in Look Files baked from Katana 1.x are not upgraded when imported with a Material node in **create from Look File** mode, or when loaded with the **asReference** option set to **true**.
- BUG ID 45963 - The Viewer proxy cache can be populated with geometry for an incorrect frame, requiring a 'Flush Caches' action to load the correct geometry.
- BUG ID 45951: Cooking, viewing, or editing a network material consisting of a network of shading nodes with many 'diamond' (rejoining) connections can result in interruptions of UI interactivity.

  This issue does not affect network materials loaded from Look Files, or otherwise imported as attributes.
- BUG ID 45942 - Node Reference in HTML format is out of date with the *Katana Reference Guide* in PDF format.
- BUG ID 45905 - The hinting mechanism for material and shading nodes is not complete.

- BUG ID 45575 - Updates performed on an active Interactive Render Filter (IRF) can sometimes activate inactive IRFs.
- BUG ID 43391 - In the GafferThree **Parameters** tab, showing incoming lights after creating local lights with colliding names puts the UI into a broken state.
- BUG ID 41877 - The shipped example asset plug-ins fail to log diagnostic messages to console.

## Other Known Issues

- BUG ID 41152 - When editing parameters of a node that is part of a LiveGroup node and reloading the parent LiveGroup node, the UI state of the **Parameters** tab is reset. This includes scroll bar positions, selections of items, and selections of nested tabs (for example **Object**, **Material,** and **Linking** tabs for a Gaffer node).
- BUG ID 41092 - When reloading a LiveGroup node's parameter interface and contents from its source, parameters of child nodes that are edited in floating panes disappear from those panes.
- BUG ID 40709 - The Alembic library does not support multiple process or thread access to an Alembic file. This means modifying an Alembic file outside Katana, while it's loaded in an open Katana scene, results in a crash. To avoid this, you must **Flush Caches** before attempting to update any modified Alembic files.
- BUG ID 40600 - Undoing a revert of an unpublished LiveGroup node does not restore the LiveGroup's editable and modified state.
- BUG ID 40599 - User parameters of non-editable LiveGroup nodes can be edited.
- BUG ID 40598 - Reverting a LiveGroup node does not revert its user parameters.
- BUG ID 40237 - Nodes can be dragged into the Group bubble of a non-editable LiveGroup node.
- BUG ID 39261 - Operations that lock and unlock nodes do not currently create entries in the Undo History, which can lead to an incorrect node graph state when undoing and redoing operations.
- BUG ID 37477 - Adding an invalid path to the ImageRead node causes Katana to crash.
- BUG ID 36926 - The **rendererSettings** > **displayOptions** parameter of a RenderOutputDefine node for the PRMan renderer (shown when its type parameter is set to raw), cannot be set using the **Parameters** tab.

  For a workaround, the UpdatePrmanSettingNodes shelf script can be used. The script, available in the PRMan17 shelf, creates an AttributeScript node that converts display options, set using string attributes, to group attributes.
- BUG ID 36691 - State badges are currently shown for attribute values of dynamic array child parameters, even though only their parent array parameter should appear with a state badge.
- BUG ID 36663 - It is not currently possible to sort notifications in the **Notifications** pop-up window.
- BUG ID 36225 - The UpdatePrmanSettingsNode shelf script does not currently deal with expression parameters. Locally set parameter expressions on PRMan nodes need re-adjusting after running the shelf script.
- BUG ID 36176 - The 2D node **Disk Render Upstream Render Outputs** option does not use the **batchRender** method for upstream render nodes, instead it currently uses **diskRender**.
- BUG ID 36170 - Control keys (notably arrow keys) do not function as expected in shell mode.
- BUG ID 34870 - Katana doesn't support render output directory creation for **shadow**, **merge,** and **script** output types.

  34256 - Previewing Rendering 2D nodes fails when **Disk Render Upstream Render Outputs** is set to **Unless Already Cached**. In addition, a crash sometimes occurs when attempting to repeat the Preview Render.
  You can work around this issue by setting the right-click context menu option on a 2D node to Always instead of

Disk Render Upstream Render Outputs to work around this problem, however, this means that each time a change is done in the 2D node and Preview Render started, a Disk Render is triggered for the image dependency upstream, even if this had no changes.

Alternatively another workaround for the 2D nodes Preview Render workflow is using an ImageRead node that is expression linked to your RenderOutputDefine node render location parameter. You need to disk render beforehand to have your image output available as 2D nodes input, and set the Monitor's 2D Render Mode to Pen-Up instead of Continuous to preview the changes as expected.

- BUG ID 33242 - Live Rendering is not designed to work with rapid updates to region of interest (ROI), and may cause unexpected behavior.
- BUG ID 31790 - In the **Parameters** tab, setting an expression for an array or a group parameter currently breaks the parameter's UI.
- BUG ID 31257 - The keyboard shortcut for **Edit** > **Toggle Ignore State of Selected Nodes** command in the **Node Graph** tab is **Alt**+**D**, not **D**. This overrides the **Alt**+**D** keyboard shortcut of the **Dim Nodes Unconnected to Viewed Node** menu item. The **D** key toggles the ignore state of a single node under the pointer.
- BUG ID 30170 - The **--crop-rect** and **setDisplayWindowToCropRect** command-line options do not currently work, so nothing is cropped.

  You can work around this issue by adding a RenderSettings node to your Katana scene with the **cropWindow** parameter set. This allows you to render using a cropped display window in batch mode.
- BUG ID 28549 - Main menu commands that create nodes, notably **File** > **Import...**, **File** > **Import LiveGroup...**, and **Help** > **I want a pony**, do not create the nodes in a Group or LiveGroup node that has been entered in a **Node Graph** tab, instead they create the nodes in the root level of the node graph document.
- BUG ID 22296 - Due to a known issue with PRMan, Katana does not currently support PRMan Live Rendering with AOVs.

  For a workaround, to allow Live Rendering, disable AOVs in interactive renders by applying an **Interactive Render Filter**, using a RenderSettings node with the **interactiveOutputs** parameter set to just **primary**.

# Release Notes for Katana 2.0v1b1

## Release Date

06 October 2014

## System Requirements

- Katana 2.0v1b1 is tested and qualified on Linux 64-bit CentOS/RHEL 5.4
- A graphics card which supports OpenGL shader model 4.0
- A supported renderer (see below)

### Supported Renderers

Katana 2.0v1b1 supports PRMan 17.0 and Arnold 4.1.3.3. The supplied renderer plug-ins are compiled and tested against these versions, using GCC 4.1.2. Minor version increments of PRMan and Arnold may work, as long as they are API compliant with the supported versions.

Support for PRMan 18.0 and above is provided directly by Pixar using their RenderMan Studio for Katana plug-in. To get access to RenderMan Studio for Katana please contact Pixar directly.

To use a version of PRMan or Arnold other than those listed above, you may need to recompile the renderer plug-in.

To expose new features and options, you may need to modify the renderer plug-in.

Using a version of PRMan or Arnold other than those listed above may produce unexpected behavior. Using a compiler other than GCC 4.1.2 may produce unexpected behavior. Please note that we can only guarantee to respond to Katana bug reports when they are reproducible with the supplied versions of the renderer plug-in, compiled with the supported version of GCC.

## Third-Party Dependencies

Katana 2.0v1b1 has dependencies on the following third-party libraries:

- OpenEXR 2.0.1
- OpenSSL 1.0.0.a

These libraries are provided in the Katana distribution, in separate directories under ${KATANA_HOME}/bin

An ABI-compatible copy of these libraries needs to reside on your LD_LIBRARY_PATH in order for many of Katana's plug-ins to run. The Katana application itself uses RPATHs to locate the required libraries.

NOTE: Katana's wrapper script ${KATANA_HOME}/katana appends ${LD_LIBRARY_PATH} to ensure these libraries are visible to Katana plug-ins.
If you manage your own LD_LIBRARY_PATH or wish to expose these libraries to plug-ins by some other means, you can call the Katana binary directly using:

```
${KATANA_ROOT}/bin/katanaBin
```

# Feature Enhancements and Important Changes for 2.0v1b1

## GafferThree

A new Gaffer node type, named GafferThree, has been created to provide improved performance when dealing with large numbers of lights in Katana projects. The GafferThree implementation takes full advantage of the new scene graph processing library Geolib3. The existing legacy Gaffer node type from Katana 1.x is still present, and current scenes should continue to work, but it is advised to move to using the new GafferThree node type where possible.

For more information, see the *What's New in Katana 2.0* document that accompanies this release.

## Graph State Variables

Node types and UI elements have been added to support the setting and querying of new, node graph-level Graph State Variables. These can be used to control which nodes in the node graph contribute to scene graph processing at a particular time. This can greatly simplify multi-pass/layer workflows or any other tasks that might want to re-use node setups with different inputs or outputs.

For more information, see the *What's New in Katana 2.0* document that accompanies this release.

## Live Rendering

- The **Live Render Control** tab has been replaced with several more Katana-native, more flexible features. These include:
  - A new column in the **Scene Graph** tab that allows toggling locations to generate Live Rendering updates when their attributes change.
  - GenericAssign-based node types named **<renderer>LiveRenderSettings** that are used to set settings for Live Rendering.
  - Qt menu actions have replaced the custom Live Rendering command buttons that were previously implemented in the renderer info plug-ins and displayed in the **Live Render Control** tab.
  - The option to use the same camera as the **Viewer** tab has been replaced by a button at the bottom of the **Viewer** tab, named **Live render from viewer camera**.

  For more information on these see the *What's New in Katana 2.0* document.
- A new LiveRenderAPI Python package provides access to several functions that allow users to modify the behavior of the Live Rendering system. These include:
  - **SendCommand()** - Sends custom Live Render commands to the render plug-in via the command socket.
  - **SendData()** - Sends custom data updates to the render plug-in via the data socket.
  - **AppendTerminalOp()** and **RemoveTerminalOp()** - Adds additional terminal Ops to the Live Rendering client allowing you to customise the scene graph data that is passed through to renderers.
  - **ClearAllTerminalOps()** - Removes all Live Rendering terminal Ops, including the default (specified in renderer info plug-ins).
  - **RestoreDefaultTerminalOps()** - Restores the default terminal Ops.
- Rebinding shaders during a PRMan Live Render now happens automatically on update without the need to manually activate from the **Live Render** menu.

## Ops

- The AttributeModifierResolve node has been deprecated in favor of OpResolve. Old scenes are still able to load and use AttributeModifierResolve nodes, but users are no longer able to create them in the UI. OpResolve does the same job as AttributeModifierResolve.
- Ops can be used to define viewer proxies on scene graph locations. Two main attribute conventions are currently supported: ViewerProxyLoader (legacy mode) and Op-based. For more information on them and examples see the *What's New in Katana 2.0* document.
- BUG ID 39971 - The options for the **applyWhen** parameter are now more consistent across the following node types:
  - AttributeFile_In
  - AttributeModifierDefine

- AttributeScript
- GenericOp
- OpScript

The following **applyWhen** options have been renamed:

- AttributeModifierDefine - **deferred** is now **during op resolve**
- AttributeScript - **during attribute modifier resolve** is now **during op resolve**
- AttributeFile_In - **deferred** is now **during op resolve** and **during Katana standard resolve** is now **during Katana look file resolve**.

## Python Tab

- Preferences for the **Python** tab in the **python** category of preferences have been revised:
  - The **commandFont** and **resultFont** preferences have been hidden, as it was not possible to change them from the **Preferences** dialog. Instead, changing the fonts can be done using the **Font** menu in the **Python** tab itself.
  - Preferences have been added to control the auto-completion behavior (**autoCompletionBehavior**) as well as whether or not to show help tooltips in the **Python** tab (**showHelpTooltips**), two features that were added in Katana 2.0v1a20.
- BUG ID 31134 - The Python tab has been improved to allow better syntax highlights, code completion, introspection info using tooltips, and block indentation and un-indentation.

## API Changes

- Attribute History: There is now an API for querying Attribute History from Python. This can be found in the `UI4.Util.AttributeHistory` module. Attribute History can be queried synchronously, in which case the UI will block until the result is computed and returned, or asynchronously if users provide a callback to run when the computation is complete. Note that displaying Attribute Inheritance is not yet implemented, but is scheduled for a later release.
- BUG ID 35076 - Katana previously shipped with a build of **OpenColorIO** that used **FnOpenColorIO** namespaced symbols, but the **OpenColorIO** libraries were not named accordingly (**libOpenColorIO.so** and **PyOpenColorIO.so** Python bindings). This caused problems with using a custom facility-installed **OpenColorIO** in parallel with the Katana libraries. This has been updated so that the libraries shipped with Katana are now Fn-prefixed too (**libFnOpenColorIO.so** and **FnPyOpenColorIO.so**). Using the Python bindings is still possible through **import FnPyOpenColorIO**
  -Any code using **PyOpenColorIO** will need updating to either use **FnPyOpenColorIO**, or a facility-installed **OpenColorIO** could be used instead.
- BUG ID 38948 - The screen window of an orthographic camera is now computed within the CameraSettings API. This calculation has been removed from the PRMan plug-in.
- BUG ID 41504 - Attributes retrieved in AttributeScript nodes using **GetAttr()** with the **asAttr** keyword argument set to **True** would be of the type **PyFnAttribute**, which is incompatible with the older **PyScenegraphAttr**. A new

bool parameter, **asFnAttribute**, was added to **GetAttr()**, **GetXform()** and **GetArguments()**, defaulting to **False**, which returns PyFnAttribute objects if set to **True**, and **PyScenegraphAttr** otherwise.

- BUG ID 43352 - The **Render** plug-in API has been renamed to **FnRender** for consistency with other Katana plug-in APIs.

## Miscellaneous

- Katana has been upgraded to use Python 2.7.3 and Qt 4.8.5, matching the VFX Platform CY2014 specification.
- Proxy caches are now considered animated by default. Static proxy caches can be defined by setting the proxies.static IntAttribute to 1.
- Curve widths, and curve knots if they exist, are now loaded into the correct attributes. Curve attributes degree and closed are now placed under the geometry group.
- The internal Katana documentation has been replaced with Sphinx-powered, searchable docs that are now opened in an external web browser. As part of this work, the **Help** tab has been deprecated, and the **Example Projects** tab has been added.
- A number nodes that were undocumented and unavailable in Katana's UI have been deprecated in Katana 2.0v1: AsmbCmpt_ PostProcess, BlockerCreate, F3D_In, GenericGeo, HDF_In, IncomingTest, NetworkAttributeScript, NetworkAttributeScriptInput, NetworkAttributeScriptNode, NetworkMaterialEdit, ProceduralSetup, Resolve, ScenegraphLocationModifierDefine, ScenegraphLocationModifierResolve, ShadowRenderAssign, TextureResolve, and TxasnMappingInfoApply.
- The VelocityApply node type has been modified to add the following two parameters:
  - **velocityAttribute** - The name of the attribute representing the velocity information to be used by the node. If the parameter is not set, the following attributes are checked:
    - **geometry.point.V**
    - **geometry.point.v**
    - **geometry.arbitrary.v**
  - **velocityUnits** - Units to be used to interpret the values stored in the velocity attribute, with the following options:
    - **units / second**
    - **units / frame**
- The way that Sky Dome items are implemented in classic Gaffer has changed. Instead of an **arnoldSurfaceShader** of type **skydome_light** on the item's Material node, materials on Sky Domes are now resolved internally in Gaffer.
- The RendererProceduralArgs node in create mode now has namespace support, similar to the namespace support already in the Material node.
- BUG ID 37514 - The **adjustScreenWindowWhen** parameter has been added to RenderSettings nodes to control when the **adjustScreenWindow** parameter is evaluated. By default, the **adjustScreenWindow** parameter is now evaluated in a deferred manner by a separate **AdjustScreenWindowResolve** Op as an implicit resolver. This allows the **renderSettings.adjustScreenWindow** attribute that is stored on the **/root** location to be modified later in downstream nodes before it takes effect. You can set **adjustScreenWindowWhen** to **immediately** to have the **adjustScreenWindow** parameter take effect immediately.

- BUG ID 38482 - A new keyboard shortcut, **Ctrl+F**, has been added to the **Node Graph**, **Parameters**, **Scene Graph**, **Attributes**, and **Project Settings** tabs to open the **Search** pop-up.
- BUG ID 40371 - The BoundsAdjust node applied bounds adjustments at all ancestor and descendent locations, rather than applying the adjustment to only the target location, and increasing any ancestor bounds to encompass the new child location bound. An **adjustAncestors** parameter has been added to enable/disable the adjusting of ancestor bounds.
- BUG ID 41293 - The node type name of Backdrop nodes has been renamed from **BackdropNote** to **Backdrop**. This is to achieve consistency with other products at The Foundry, such as NUKE.
- BUG ID 41366 - The Alembic_In node now supports a **useOnlyShutterOpenCloseTimes** parameter that forces the Alembic cache to only use the time samples corresponding to shutter open and close times when the **maxTimeSamples** parameter is set to **2**. The parameter is available in the **advanced** section in the **Parameters** tab. Note that the useOnlyShutterOpenCloseTimes argument is also supported by the AlembicIn Op.
- BUG ID 42439 - The pixel probe toolbar in the **Monitor** tab has been modified to show a luminance component for sampled pixels next to the existing RGBA components. Also, the menu buttons for selecting a scene graph location in the **Scene Graph** tab and for copying the path of a scene graph location are now only shown when a scene graph location corresponds to the object under the pointer, as determined through an ID pass, which can be turned on using the **Render** > **3D Rendering** > **Render ID Pass** toggled menu item.
- BUG ID 43300 - The **imageInfo** parameter passed to **FnRenderOutputLocationPlugin::computeFilePath()** now includes the following new fields:
  - **scenePassName** - a string including the scene file name and the Render node's **passName** parameter's value, for example, mySceneFileName_BeautyPass. Notice that this is the same value set for **passName**.
  - **renderPassName** - the Render node's **passName** parameter's value, for example, BeautyPass.
  - **sceneName** - the scene file name, for example, mySceneFileName.
- BUG ID 43302 - RenderOutputLocation plug-in APIs have been extended to allow the **FnRenderOutputLocationPlugin::computeFilePath()** and **FnRenderOutputLocationPlugin::computeLocation()** methods to access the renderSettings group attribute stored on the /root location of the scene graph.

  Methods' signature has changed as shown below:

```
virtual std::string computeFilePath(
const FnAttribute::GroupAttribute & outputAttrs,
  const FnAttribute::GroupAttribute & locationAttrs,
  const FnAttribute::GroupAttribute & renderSettingsAttrs,
  const FnAttribute::GroupAttribute & imageInfo,
  bool makeVersionTemplate) = 0;
virtual std::string computeLocation(
  const FnAttribute::GroupAttribute & outputAttrs,
  const FnAttribute::GroupAttribute & locationAttrs,
  const FnAttribute::GroupAttribute & renderSettingsAttrs,
  const FnAttribute::GroupAttribute & imageInfo) = 0;
```

  Custom render output location plug-ins have to be recompiled in order to be compatible with the new signatures.

- BUG ID 43526 - PRMan coshader parameters with the coshaderPort hint enabled did not correctly appear in the connectable list.
- BUG ID 43632 - Asset IDs in Viewer proxies were not being resolved.
- BUG ID 44555 - The channel name provided to the renderer-specific OutputChannelDefine nodes may now contain dot (**.**) characters. This is supported through the addition of an exact **name** child parameter of the corresponding, safely-named group attribute under [renderer]**GlobalSettings.outputChannels**.

# Bug Fixes

- ImageChannels nodes were missing the multi-input port in the **Node Graph** tab.
- Katana's crash handler did not handle SIGABRT, SIGILL or SIGFPE signals, but rather only SIGSEGV.
- The DependencyMerge node type used legacy names for the **farmSettings** parameter, and the **farmSettings.farmFileName** parameter did not appear correctly for Render nodes.
- BUG ID 27147 - Due to an internal bug in Qt 4.7.2, an error message similar to the following was occasionally displayed in the console: "Application asked to unregister timer 0x13000076 which is not registered in this thread. Fix application." (https://bugreports.qt-project.org/browse/QTBUG-16175).
- BUG ID 28477 - When editing the parameters of a Gaffer node in the **Parameters** tab and using the mouse wheel to scroll with the pointer over the title of a nested tab (**Object**, **Material**, or **Linking**), the tabs under the pointer were switched while also scrolling widgets in the **Parameters** tab in the direction of the wheel. The new behavior is to not switch tabs when using the mouse wheel in the **Parameters** tab, which is consistent with behavior in Nuke.
- BUG ID 29524 - Color swatches in the GafferThree table were not shown with Filmlook Visualization/Display Transform applied.
- BUG ID 32062 - The **Edit Expression...** command in the context menu of parameters in the **Parameters** tab was incorrectly placed between the **Expression** and **Curve** menu items.
- BUG ID 33784 - The LightCreate node's **makeInteractive** parameter was not hinted as boolean.
- BUG ID 35672 - An exception was raised after choosing the '**Find and Select'Selected Items** option from the right click menu for a CEL collection in the **Parameters** Tab.
- BUG ID 36422 - The **Messages** tab, pop-up window, and indicator button suffered from poor performance when logging a large number of messages through Python.
- BUG ID 36428 - When pressing certain keyboard shortcuts over some tab types, such as the Tab key over the **Node Graph** or **Monitor** tabs while a modal dialog or pop-up widget were shown, the keyboard shortcuts were incorrectly still attached to widgets underneath the cursor, ignoring the modal state of the dialog or pop-up widget.
- BUG ID 36687 - Setting a key frame on a parameter was not indicated by a green background.
- BUG ID 36700 - Local renders failed when the host name of an artist's workstation could not be resolved. Katana now falls back on 'localhost'.
- BUG ID 38307 - Calling **NodegraphAPI.CreateNode()** from the Python API to create a node created multiple entries in the undo history.

- BUG ID 38752 - Terminating a Katana UI session with a SIGTERM or SIGINT signal during a render didn't properly terminate the render process that continued holding a license.
- BUG ID 38839 - Orthographic camera projection in the Viewer was incorrect.
- BUG ID 40375 - Setting the **crop** parameter of a ZoomToRect node to **Yes** did not have an effect on the rendered image.
- BUG ID 40404 - Katana could become unresponsive for about 20 seconds when enabling a KVM switch.
- BUG ID 40414 - Viewer Modifier Plug-in API: the **setAttributeOverride()** and **setLiveAttributeOverride()** functions have been deprecated.
  Since only one VMP may run on any scene graph location, there is no need to provide overrides outside of the individual plug-in.
- BUG ID 41072 - Client code was required to explicitly declare **errorMessage** as an attribute in SceneGraphGenerators as it was automatically queried if location type was **error**.
- BUG ID 41117 - The **Last File Save** bookmark was not written to disk when a project was first saved.
- BUG ID 41236 - The ShadowManager uses a temporary directory for storing shadow renders, but would only evaluate **KATANA_TMPDIR** when a light was added to a pass, resulting in erroneous render output locations when the scene was reopened.
- BUG ID 41240 - The **Export Scene Graph Bookmarks** dialog did not prompt before overwriting files.
- BUG ID 41584 - Errors in registering plug-ins for the Gaffer node caused Katana to fail to load.
- BUG ID 41586 - The shortcut **Ctrl**+middle-mouse button+drag and drop from the **Attribute** tab into the **Python** tab produced an explicit PyFnAttribute representation. If required **Ctrl**+**Shift**+middle-mouse button generates the old PyScenegraphAttr code.
- BUG ID 42117 - The **sss_faceset** parameter was deprecated from ArnoldObjectSettings for Arnold 4.1 since it is no longer supported by Arnold 4.1.
- BUG ID 42132 - The Viewer required a geometry.knots attribute when drawing curves, even though that attribute and its contents were neither needed nor used.
- BUG ID 42168 - In Katana 1.x the attribute convention for referenced materials used **reference.lookfile**, now it uses **reference.asset**. Material Resolve (both the Op and DAP) now also checks for the old convention in order to handle, for example, a baked scene containing **reference.lookfile**. The nodes that create a referenced material location (LookFileMaterialIn, for example) will follow the new convention.
- BUG ID 42201 - An exception was raised when middle-click dragging a cell in the **Color** column in a row of a light of a Gaffer node that was edited in the **Parameters** tab.
- BUG ID 42463 - The Importomatic CastingSheet.py example's XML tag was named incorrectly. The correct form is <castingsheet>.
- BUG ID 42698 - The functions **pystring.isupper()** and **pystring.islower()** only returned **true** if the input string was set so that every character was of the correct case. Now these functions return **true** if every cased character in the input string is of the correct case. This new behavior matches Python's definition of these functions.
- BUG ID 42738 - GafferThree: exported rigs used the **.gprig** extension, which is used by the Gaffer node, instead of the **.rig** extension.
- BUG ID 43449 - When certain text fields had focus, keyboard shortcuts such as **Ctrl**+**O** and **Ctrl**+**N**, did not work.

- BUG ID 43467 - Attempting to get the default value of an array parameter through a GenericAssignParameterPolicy instance, returned a **None** value.
- BUG ID 43580 - The 'hide' hint specified in an args file was ignored for coshader ports on shading nodes.
- BUG ID 44238 - The **SocketIdSenderException** raised in the FnRender API by **SocketIdSender** wasn't a subclass of **std::exception**, and didn't provide any description of an error when thrown.
- BUG ID 44257 - PRMan Live Rendering: The render camera occasionally wasn’t set correctly on render start, causing camera updates to be ignored.
- BUG ID 44328 - An editable PopupFormWidget, when clicked in certain regions of the widget, received keyboard focus but not keyboard input, even after additional mouse clicks.
- BUG ID 43525 - If a coshader parameter enables the coshaderPort hint and has a label hint defined, the label is now used in the connectable port list instead of the parameter name.

  This is only applied when a nodeType is set or changed. Existing nodes still show the parameter name.
- BUG ID 44852 - An **includeInCameraList** parameter incorrectly appeared in the parameter interface of LightCreate nodes. The parameter is now hidden.
- BUG ID 45364 - The provided Alembic_In example SGG Makefile added Alembic libraries in an incorrect order.